

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ СОЗДАНИЯ И
ПРОХОЖДЕНИЯ ПСИХОЛОГИЧЕСКИХ ТЕСТОВ С
ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМЫ ASP.NET**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Ходаковской Варвары Андреевны

Научный руководитель
старший преподаватель

М.В. Белоконь

Заведующий кафедрой
к.ф.-м.н., доцент

Л.Б. Тяпаев

Саратов 2024

ВВЕДЕНИЕ

В современном мире роль технологий в психологических исследованиях становится все более значимой. Создание web-ресурса для психологических исследований особенно актуально, так как сайты это одни из самых удобных инструментов для проведения психологических тестов. Они позволяют людям получать доступ к психологическим тестам в любое время и из любого места.

Данная работа посвящена разработке web-приложения, предназначенного для создания и редактирования тестов, прохождения онлайн тестирования, а также отслеживания результатов и получения статистики. Сайт будет написан с использованием технологии ASP.NET.

ASP.NET – это платформа для создания web-приложений, разработанная компанией Microsoft в 2000 г. Основные преимущества ASP.NET – это гибкость, высокая производительность, безопасность.

Целью дипломной работы является разработка web-приложения для создания и прохождения психологических тестов с использованием платформы ASP.NET.

Для достижения цели работы были поставлены следующие задачи:

- Изучение платформы ASP.NET и ее основных принципов работы.
- Определение необходимых функций и возможностей приложения.
- Проектирование структуры и логики web-приложения.
- Проектирование базы данных на основе SQLite.
- Реализация серверной логики на языке C#.
- Обеспечение надежной защиты сайта.
- Разработка пользовательского интерфейса.
- Тестирование и оптимизация приложения.
- Загрузка приложения на web-хостинг.

Бакалаврская работа состоит из введения, 3 разделов, а именно: «Психологические тесты», «Выбор инструментов разработки», «Проектирование сайта», заключения, списка использованных источников и приложения доступного на USB-флеш-накопителе. Общий объем работы – 49 страниц, из них 49 страниц – основное содержание, включая 49 рисунков, список использованных источников – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первом разделе «Психологические тесты» представлены общие термины, связанные с психологическими тестами, рассказана история появления тестов, рассмотрены различные типы тестов.

Психологическое тестирование – термин психологии, обозначающий процедуру установления и измерения индивидуально-психологических отличий.

Психологические тесты начали активно развиваться в XIX веке и стали популярными во времена Первой мировой войны. Психологическое тестирование используется для измерения индивидуально-психологических отличий. Тесты помогают оценить уровень стресса, тип личности, черты характера, причины проблем с окружающими и многое другое.

В психодиагностике выделяют несколько методов, такие как диагностические методы, вербальные и невербальные методы, и методические принципы классификации, такие как объективные тесты, стандартизированные самоотчеты, проективные техники и другие.

В разрабатываемом приложении можно будет создавать тесты на определение личности, мотивации, темперамента, тесты ТЮФ и т. д.

Во втором разделе «Выбор инструментов разработки» описана история создания и развития платформы ASP.NET, модели семейства ASP.NET, особенности платформы. Особое внимание уделено ASP.NET MVC, подробно рассказано про концепцию Model-View-Controller. Также описаны основные инструменты для работы с данными – база данных SQLite, ORM Entity Framework.

В 1996 году появилась технология Active Server Pages (ASP). Она представляла собой платформу для создания динамических web-страниц, которые могли взаимодействовать с базами данных и другими серверными ресурсами.

В 2002 году была представлена новая технология ASP.NET входящая в состав пакета .NET Framework 1.0. Несмотря на то, что внешне она была схожа с предыдущей технологией, внутреннее устройство ASP.NET существенно отличалось от ASP. Новая технология основана на платформе .NET и, следовательно, использует все возможности, предоставляемые этой платформой. В 2009 году Microsoft представила еще одну модель разработки – ASP.NET MVC.

В семействе ASP.NET есть и другие модели, такие как ASP.NET Web API для создания веб-служб, ASP.NET SignalR для добавления веб-функций в реальном времени и ASP.NET Blazor для создания интерактивных веб-приложений с

использованием C# и .NET.

ASP.NET обладает рядом преимуществ, включая интеграцию с .NET Framework, поддержку множества языков программирования, объектно-ориентированный подход, легкость развёртывания и конфигурирования приложений.

Для создания web-сайта в рамках дипломного проекта была выбрана модель ASP.NET Core MVC.

ASP.NET MVC (Model-View-Controller) – это фреймворк для создания web-приложений, разработанный компанией Microsoft. Он представляет собой реализацию архитектурного паттерна MVC, который предполагает разделение приложения на три компонента:

- **Модель (model)** описывает используемые в приложении данные, а также логику, которая связана непосредственно с данными, например, логику валидации данных. В то же время модель не должна содержать логику взаимодействия с пользователем и не должна определять механизм обработки запроса. Кроме того, модель не должна содержать логику отображения данных в представлении. Как правило, объекты моделей хранятся в базе данных.
- **Представление (view)** отвечает за отображение данных пользователю. Это HTML-шаблоны, в которых отображается информация из модели. Представления являются ответом сервера на запросы клиента и отображают данные в удобном для пользователя формате. Также представление может содержать логику, связанную с отображением данных. В то же время представление не должно содержать логику обработки запроса пользователя или управления данными.
- **Контроллер (controller)** является некой «прослойкой» между моделью и представлением. Он является «сердцем» приложения так как контроллер обрабатывает запросы от клиента, взаимодействует с моделью для получения данных и передает эти данные представлениям для отображения. Хранение данных будет реализовано с помощью базы данных SQLite.

SQLite – это компактная встраиваемая реляционная база данных, она обеспечивает хранение данных без необходимости отдельного сервера.

Взаимодействие с базой данных SQLite будет производиться через специальный ORM – Entity Framework.

Entity Framework (EF) – это объектно-ориентированный ORM (Object-

Relational Mapping) фреймворк, разработанный Microsoft для работы с базами данных в приложениях .NET. Он предоставляет собой удобный способ взаимодействия с базами данных через объектно-ориентированный подход, позволяя разработчикам работать с данными, как с обычными объектами в коде. Отличительной чертой Entity Framework является использование LINQ.

В третьем разделе «Проектирование сайта» описаны требования к сайту, процесс проектирования базы данных и подключения, основные этапы программной реализации web-приложения, рассказано про систему маршрутизации, про создание моделей, контроллеров и представлений. Особое внимание было уделено безопасности сайта, описан процесс хеширования паролей, аутентификации и авторизации с использованием JWT-токенов. Также подробно рассмотрен интерфейс приложения и выгрузка его на хостинг.

Для успешного функционирования сайта, были выделены следующие требования:

1. Web-приложение должно иметь два личных кабинета – администратора и студента:
 - В личном кабинете администратора должны быть функции регистрации и авторизации для доступа к управлению тестами. Функционал должен включать в себя добавление новых тестов, вопросов и ответов, создание категорий и суждений. Администратор также должен иметь возможность редактировать уже существующие тесты и просматривать статистику результатов тестирования студентов с возможностью фильтрации по различным параметрам, таким как студент, группа, дата и т.д.
 - В личном кабинете студента должны быть функции регистрации и авторизации. А также студент должен иметь возможность просматривать доступные тесты, их описание, проходить тестирование, просматривать свои результаты. Важно предусмотреть возможность формирования статистики по результатам прохождения тестов для каждого студента.
2. Система должна поддерживать два основных типа тестов:
 - Тесты с вариантами ответов и оценкой по баллам – эти тесты предполагают выбор одного или нескольких вариантов ответов на вопросы. Каждый ответ оценивается определенным количеством баллов.
 - Тесты с суждениями и категориями (альтернатива раскладывания карточек по кучкам) – в этих тестах студенту необходимо каждому суждению

сопоставить определенную категорию. Ответы не оцениваются, но используются для анализа.

3. Помимо общедоступных тестов, должна быть возможность создавать тесты, доступные только по паролю. Пароли для таких тестов могут быть заданы администратором или сгенерированы автоматически системой.
4. Интерфейс сайта должен быть удобным и интуитивно понятным как для администратора, так и для студентов. Для удобства использования важно предусмотреть простую навигацию, понятные инструкции и информативные сообщения об ошибках.
5. Сайт должен обеспечивать безопасность данных, то есть защищать от несанкционированного доступа, SQL-инъекций, утечки данных.

База данных содержит восемь таблиц: «Администратор», «Типы категорий», «Категории», «Вопросы», «Ответы», «Студент», «Статистика тестов», «Статистика тестов второго типа».

Для разработки ASP.NET MVC приложения был создан новый проект в Visual Studio с использованием шаблона «Веб-приложение ASP.NET Core (модель-представление-контроллер)». Далее, необходимо было подключить базу данных, для этого была создана строка подключения, и настроен контекст данных.

Следующим этапом было необходимо создать модели, которые содержат ключи и свойства, соответствующие столбцам таблиц в базе данных. Также были созданы несколько дополнительных моделей, нужных для реализаций функций приложения, но данные из этих классов (моделей) не требуют хранения в базе данных.

В рамках паттерна MVC контроллеры играют ключевую роль в обработке запросов, поступающих от клиентов. Методы контроллера вызываются из web-среды через некоторый URL для выполнения какого-либо действия. Например, в приложении создано 8 контроллеров, каждый из которых содержит методы для работы с соответствующей моделью. Каждый метод соответствует одному из типов HTTP-запросов.

Работа контроллера заключается в подготовке данных и передаче их представлению, которое отвечает за их визуализацию в виде HTML-разметки. Для создания клиентской части приложения были использованы такие языки инструменты как: CSS, HTML, JavaScript, Bootstrap.

Чтобы предотвратить быстрое раскрытие паролей злоумышленниками в случае утечки базы данных, важно хранить пароли в неявном виде. Для шифрования паролей использовано хеширование в сочетании с «солью», при помощи библиотеки BCrypt.NET – алгоритм `HashPassword()`. Для настройки аутентификации (проверка подлинности) и авторизации были использованы два специальных JWT токена, схема аутентификации `JwtBearerDefaults`. Первый токен необходим для авторизации, второй – для обновления пары токенов: токен доступа (`Access token`) и токен обновления (`Refresh token`). Для защиты сайта от SQL-инъекций использовался `Entity Framework`.

Интерфейс приложения разработан таким образом, чтобы он был интуитивно понятен. Web-приложение состоит из 26 представлений – главная страница с кнопками входа администратора и студента, кабинет администратора с возможностью добавления и редактирования тестов, вопросов и ответов к тесту, просмотра статистики по двум типам тестов и подробной информации (с графиками) о прохождении каждого теста, в кабинете студента реализована возможность выбора теста и его прохождения, просмотр результатов.

После успешного тестирования и рефакторинга приложение было выложено на хостинг `deploy-f`.

ЗАКЛЮЧЕНИЕ

В рамках данной бакалаврской работы была выполнена разработка web-приложения для создания и прохождения психологических тестов с использованием платформы ASP.NET. Цель работы была достигнута посредством выполнения ряда поставленных задач, включающих изучение платформы, проектирование структуры и логики приложения, создание базы данных и написание программного кода, а также обеспечение защиты и оптимизация приложения.

В ходе работы была проведена детальная проработка возможностей и функций, необходимых для реализации web-приложения. Проектирование структуры и логики позволило создать устойчивую архитектуру приложения, а использование базы данных SQLite обеспечило надежное хранение и управление данными пользователей и тестов.

Серверная часть была написана на языке C#, клиентская часть – HTML, CSS, JavaScript, Bootstrap, что позволило эффективно реализовать функционал приложения, включая создание и прохождение тестов, отслеживание статистики и администрирование контента. Особое внимание было уделено обеспечению безопасности приложения, средствам шифрования данных, использования Access/Refresh токенов, ORM.

После завершения всех этапов разработки, приложение было успешно загружено на web-хостинг, что позволило пользователям использовать его в реальном времени. В результате проделанной работы, созданное web-приложение полностью соответствует требованиям и обладает всеми необходимыми функциями для создания и прохождения психологических тестов.

Таким образом, поставленная цель достигнута, задачи полностью выполнены.

Основные источники информации:

- 1 Карелин А. Большая энциклопедия психологических тестов. – Москва: Эксмо, 2007. – 416 с.
- 2 Бурлачук Л. Ф. Психодиагностика: учебник для ВУЗов. – СПб.: Питер, 2006. – 351 с.
- 3 Фримен А. ASP.NET MVC 5 с примерами на C# 5.0 для профессионалов. – Москва: Вильямс, 2014. – 736 с.
- 4 Путь ASP.NET Core [Электронный ресурс] URL: <https://habr.com/ru/articles/312226/> (Дата обращения – 10.02.2024) – Загл. с экрана. – Яз. рус.

- 5 ASP.NET MVC Pattern [Электронный ресурс] URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc> (Дата обращения – 25.02.2024) – Загл. с экрана. – Яз. англ.
- 6 SQLite [Электронный ресурс] URL: <https://blog.skillfactory.ru/glossary/sqlite/> (Дата обращения – 07.03.2024) – Загл. с экрана. – Яз. рус.
- 7 Entity Framework Core [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/ef/core/> (Дата обращения – 25.04.2024) – Загл. с экрана. – Яз. англ.